The Taligent Application Environment CommonPoint™

Presented by: Peg MacPhail
IBM Austin, Tx.

The Taligent Proposition

Topics

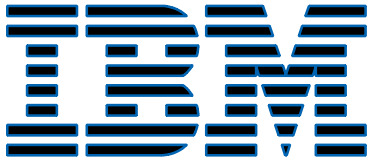
- The Taligent Company
  - Company background
  - Architectural approach
  - Products
  - Product staging

The Taligent Proposition

- Incorporated March 2, 1992 to
  - Deliver the next generation of system software
  - Encourage innovation and differentiation
  - Provide a high volume application platform

- Small/Med. Bus.
- "Pink" Technology
- EU Innovation

- Enterprise Position
- Dist. OO Technology
- Open Standards



- Enterprise Leader
- Adv. Technologies
- Market Position

Industry Challenges

Commercial ISVs
- Large, feature–laden apps
- Fewer vertical/innovative apps
- Development cost is a barrier
- >2 years to develop

Hardware Manufacturers
- Little product differentiation
- Innovation restricted by legacy OS
- Commodity market – low margins

Corporate Developers
- Inability to quickly develop apps to capture business processes
- Lack of vertical or domain–specific applications
- Maintenance burden as high as 80% of IS resources
- Lack of modular, flexible applications

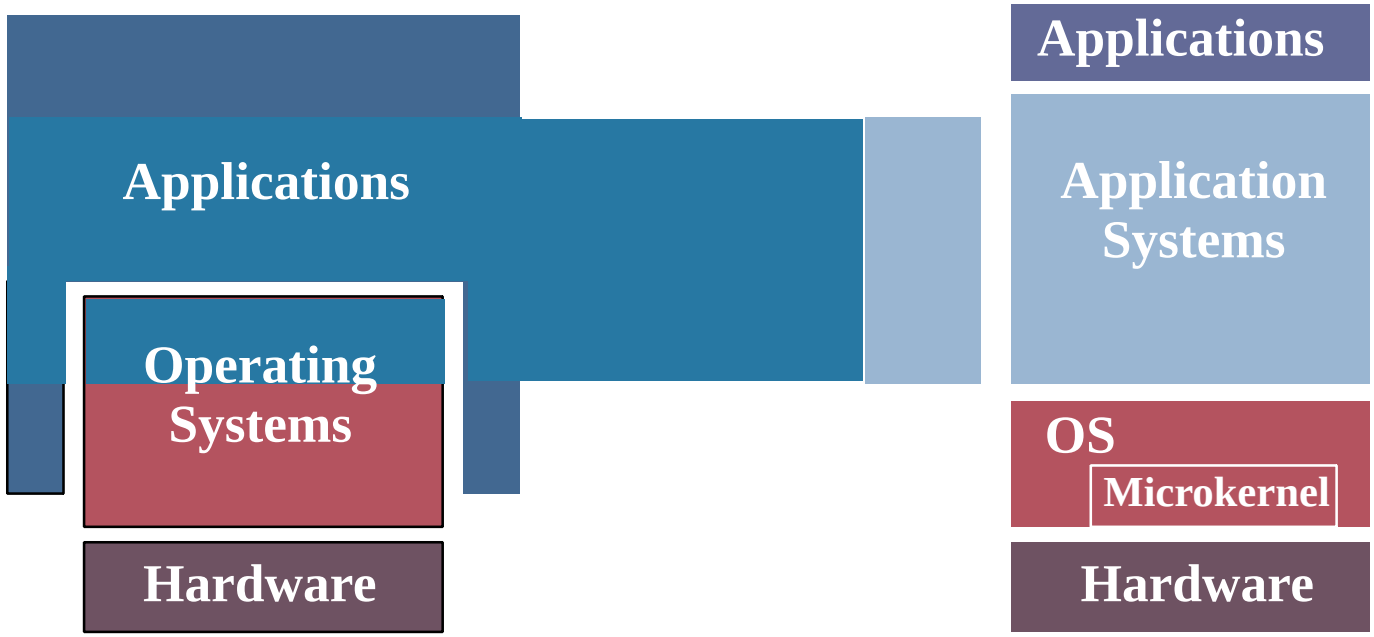
Users
- Forced to learn complex computer and network navigation methods
- Applications lag the business opportunities
- Existing groupware forces adoption of un–natural work processes

The Taligent Proposition

- **Business objectives**
  - **Enable dramatic improvements in developer productivity**
  - **Business critical computing focus**
  - **High volume deployment**
  - **Encourage the development of a new class of applications**

Approach . . . fully leverage object technology

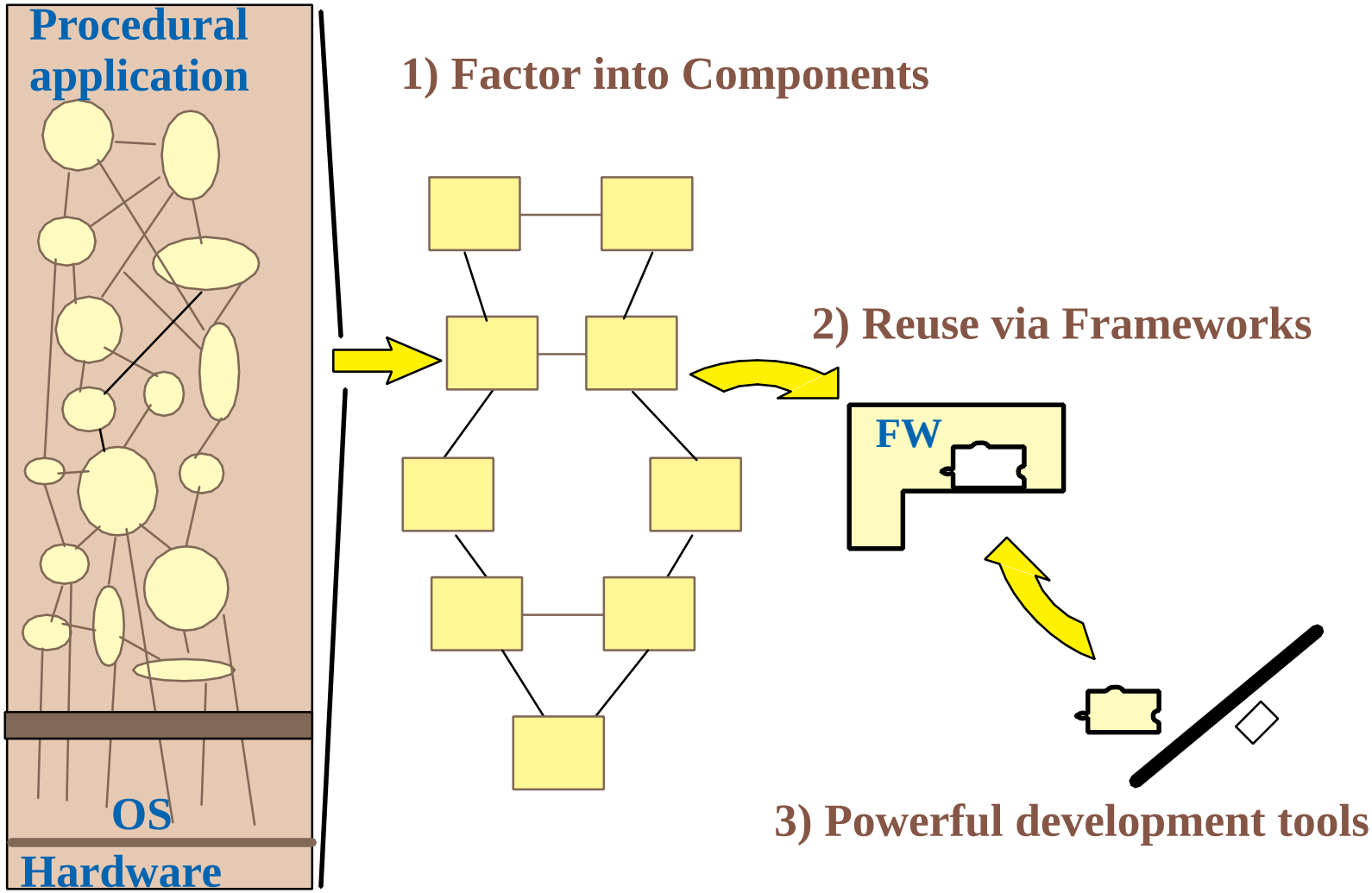
Application System Emerges



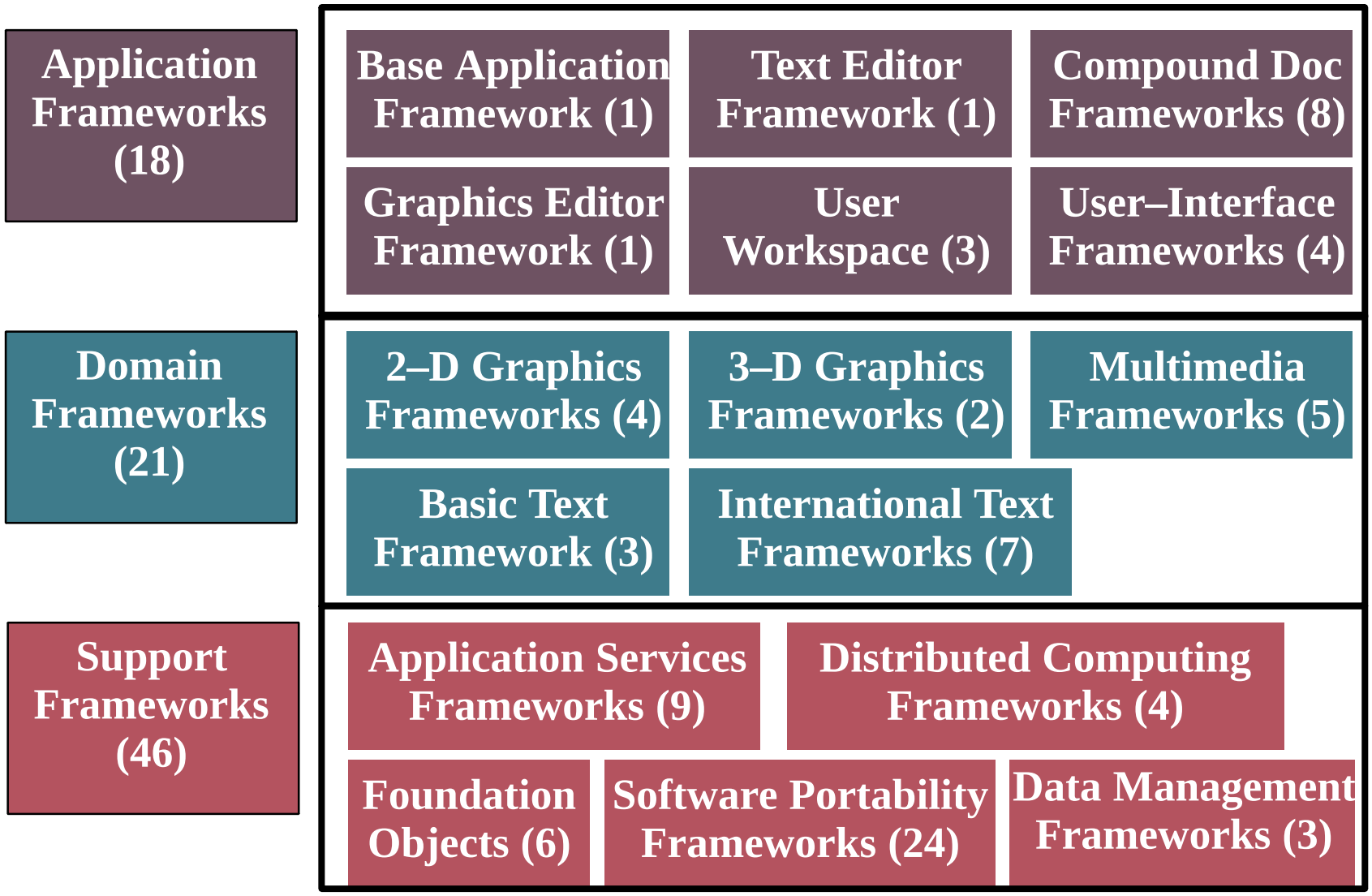
Early ’90s

Mid ’90s & Beyond

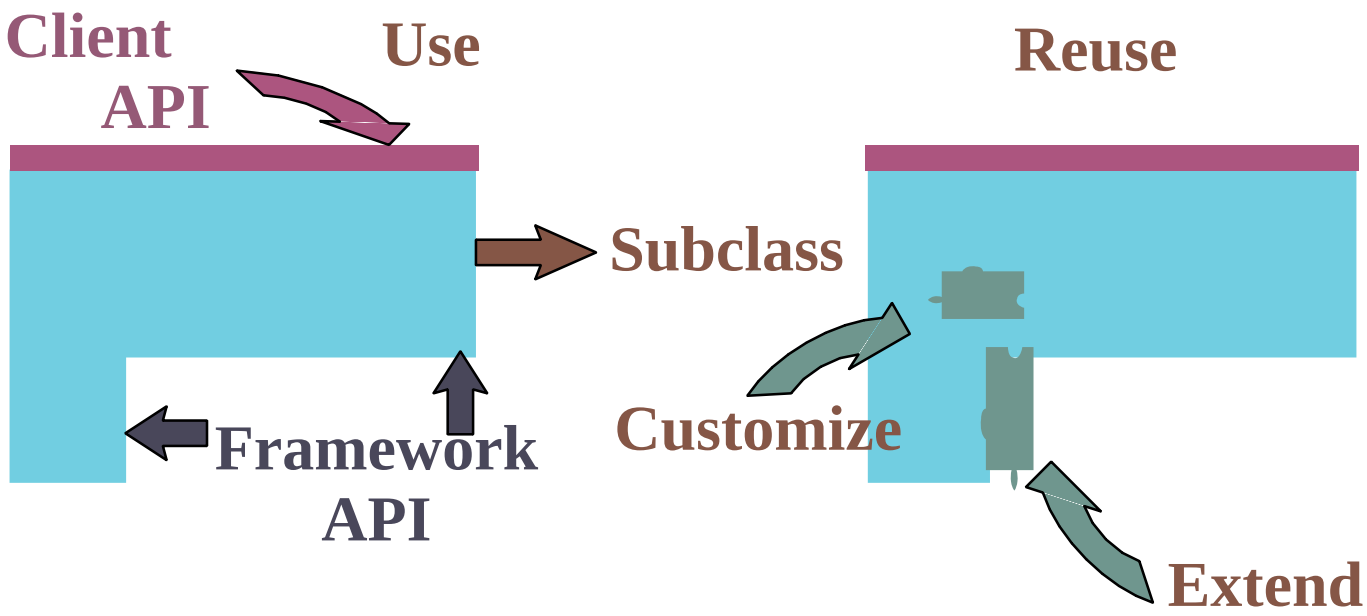
Taligent Architecture



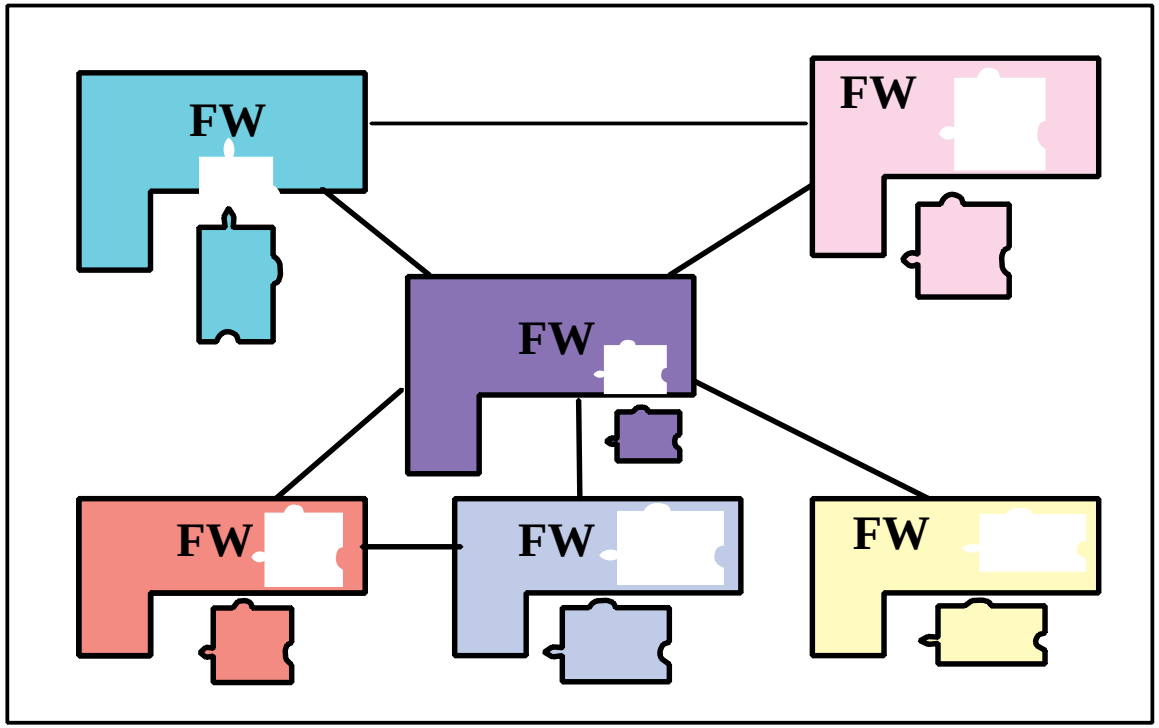
CommonPoint™: Architecture Overview



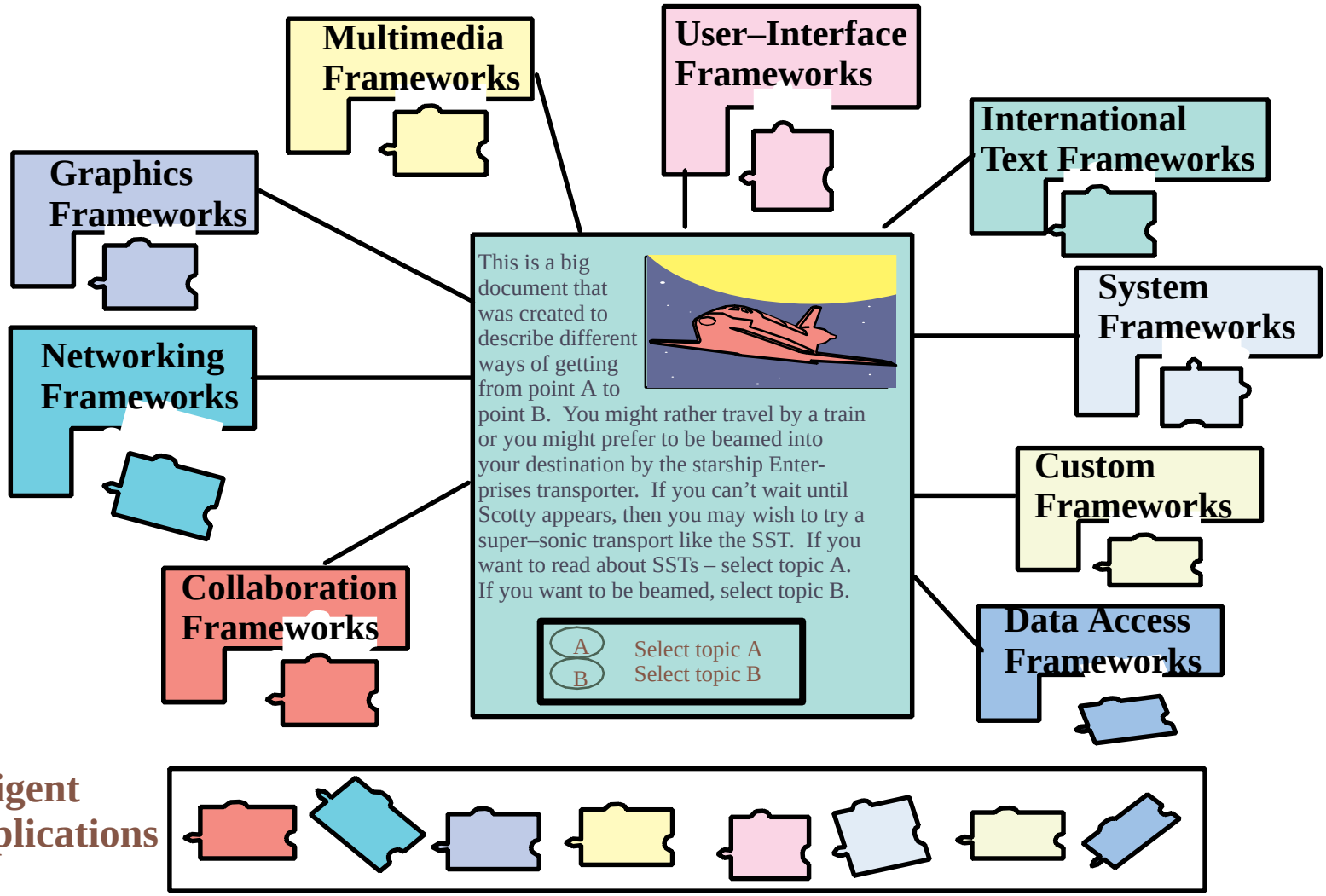
Framework


Client
API
Use
Reuse
Subclass
Framework
API
Customize
Extend

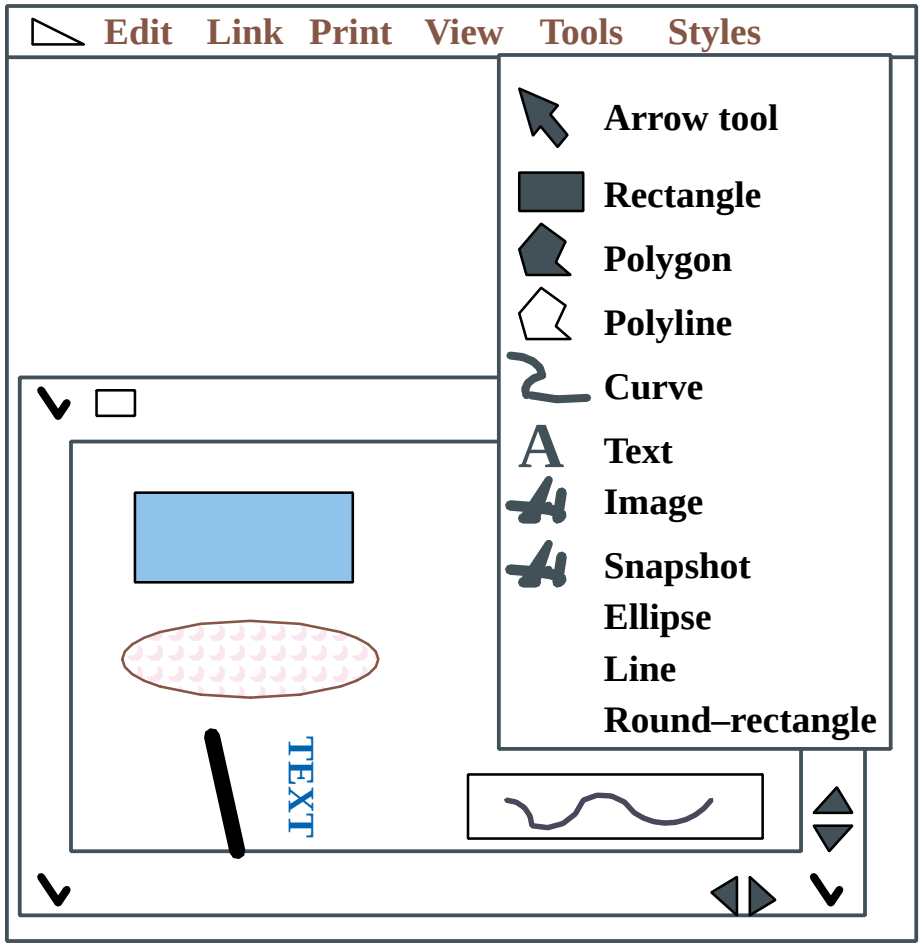
Building on Frameworks



Extensive Functionality . . . with Reuse


Multimedia Frameworks
User–Interface Frameworks
International Text Frameworks
Graphics Frameworks
System Frameworks
Networking Frameworks
Custom Frameworks
Collaboration Frameworks
Data Access Frameworks
This is a big document that was created to describe different ways of getting from point A to point B. You might rather travel by a train or you might prefer to be beamed into your destination by the starship Enterprises transporter. If you can't wait until Scotty appears, then you may wish to try a super–sonic transport like the SST. If you want to read about SSTs – select topic A. If you want to be beamed, select topic B.
A
B
Select topic A
Select topic B
Taligent Applications

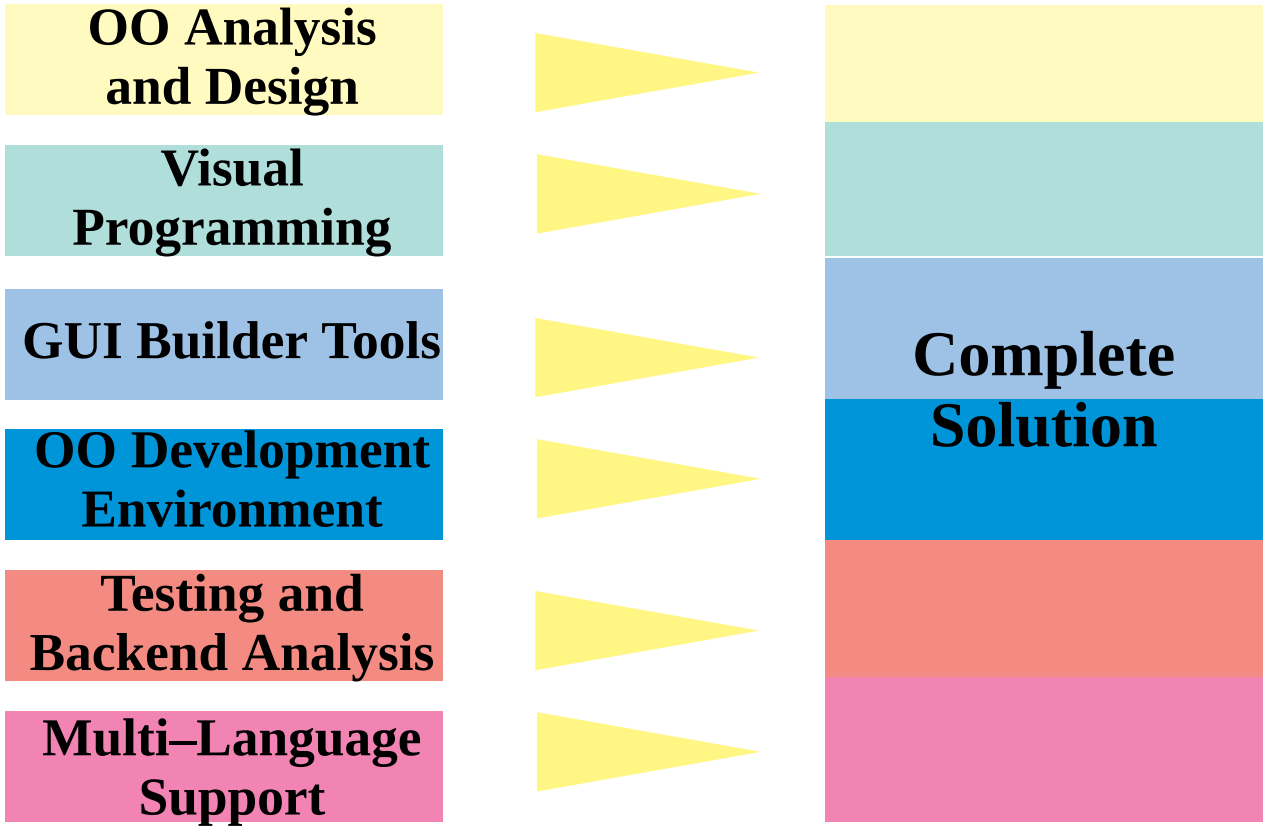
Graphical Editing Framework



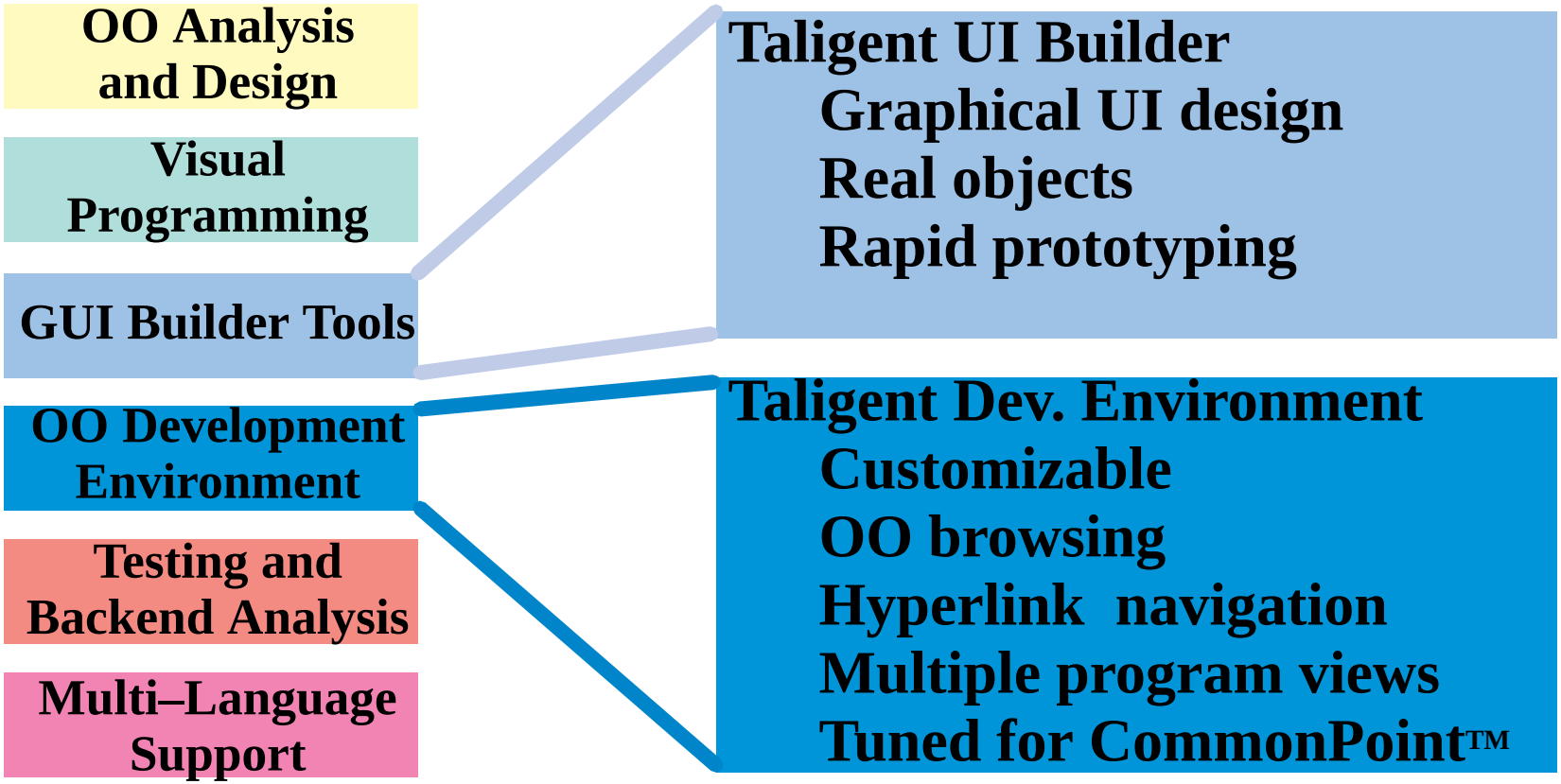
- Direct Manipulation
- Draw
- Select
- Translate, Scale, Rotate
- Group
- Connect

- Plus, Inherit from Doc Fwk:
  - Printing
  - Filing, Saveless Model
  - Copy, Paste
  - Embed, Link
  - Multilevel Undo
  - Scripting
  - Collaboration
- Plus, Inherit UI Fwk:
  - Windows, menus
  - Event processing
  - Auto scrolling
  - Drag & drop

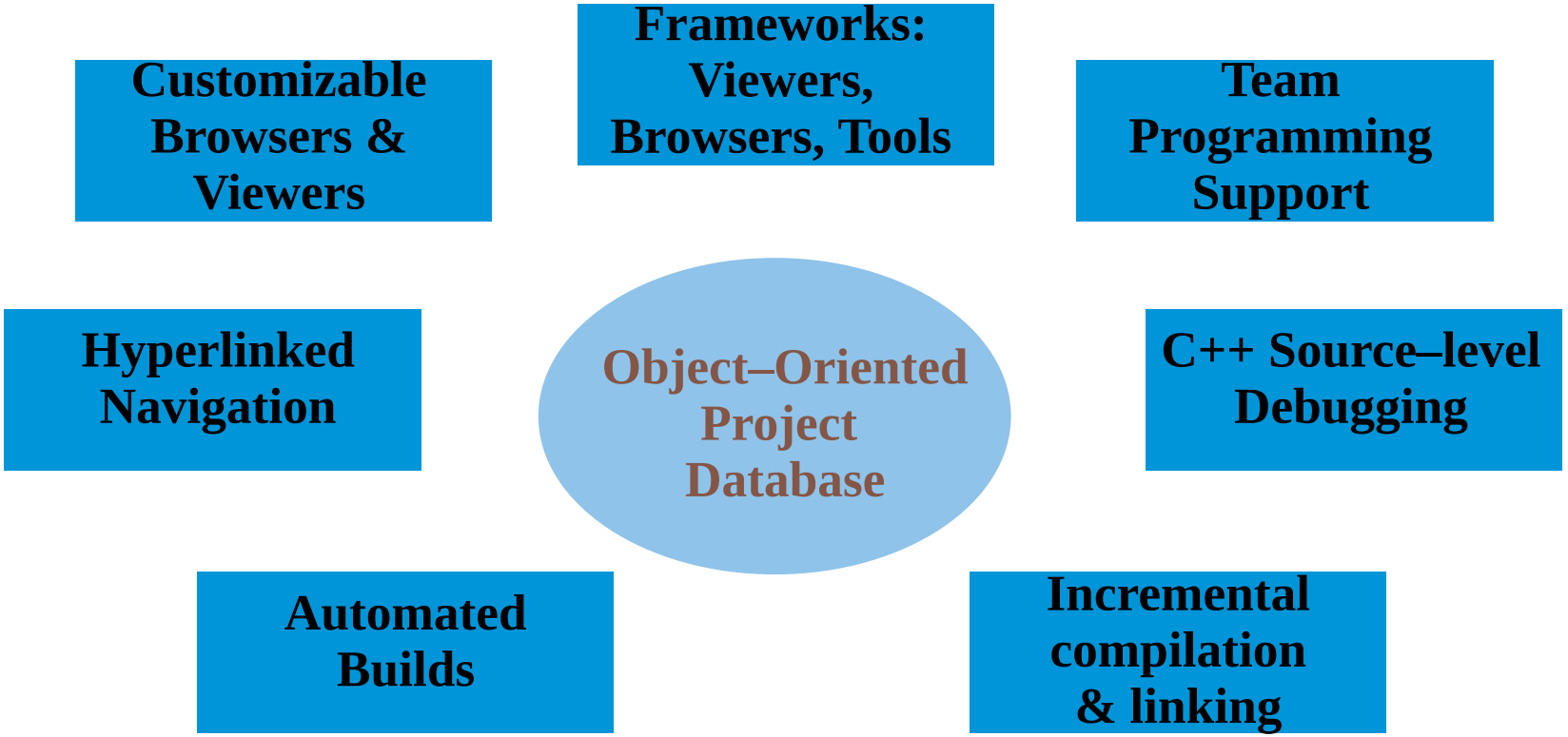
OO Development Tools Spectrum



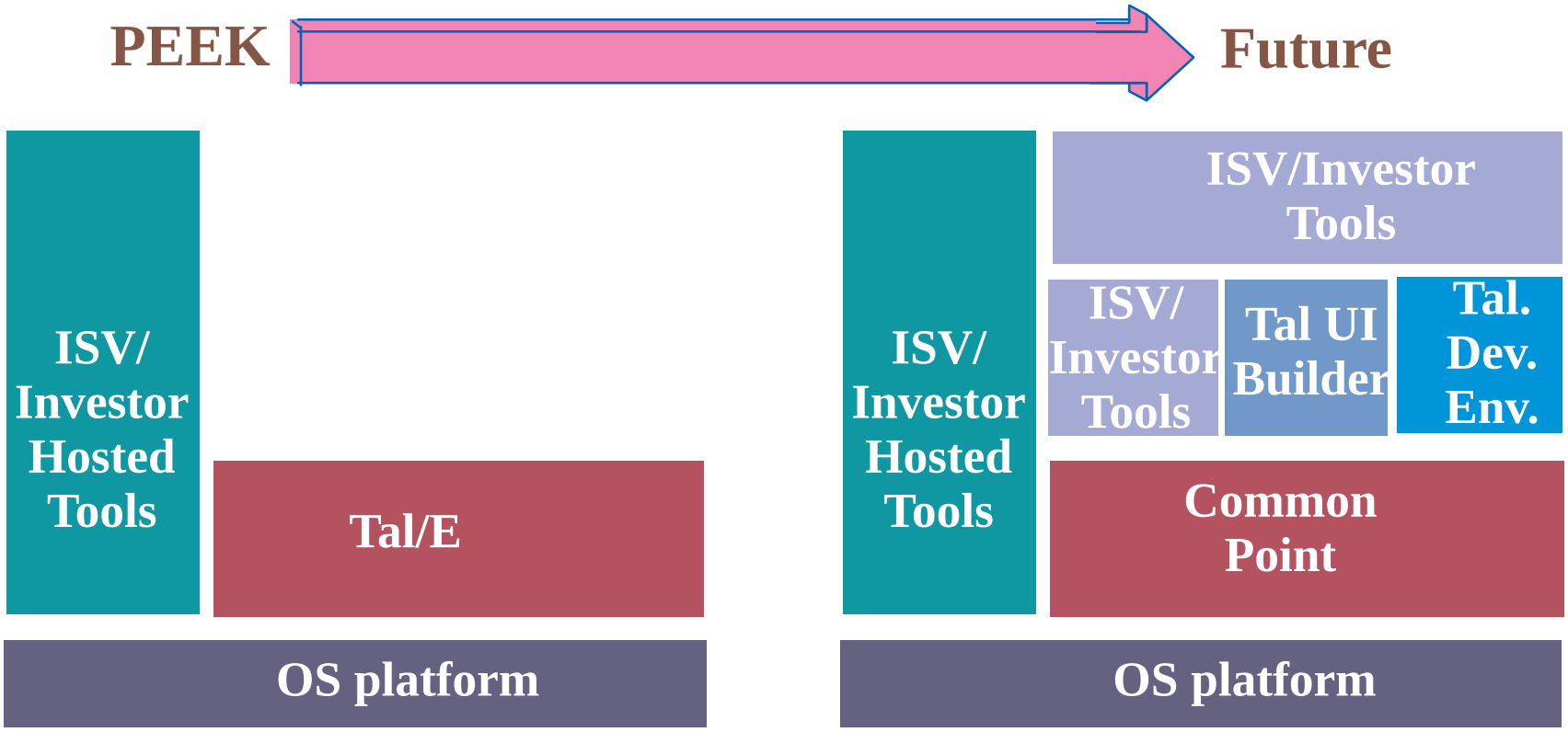
Taligent Tool Offerings



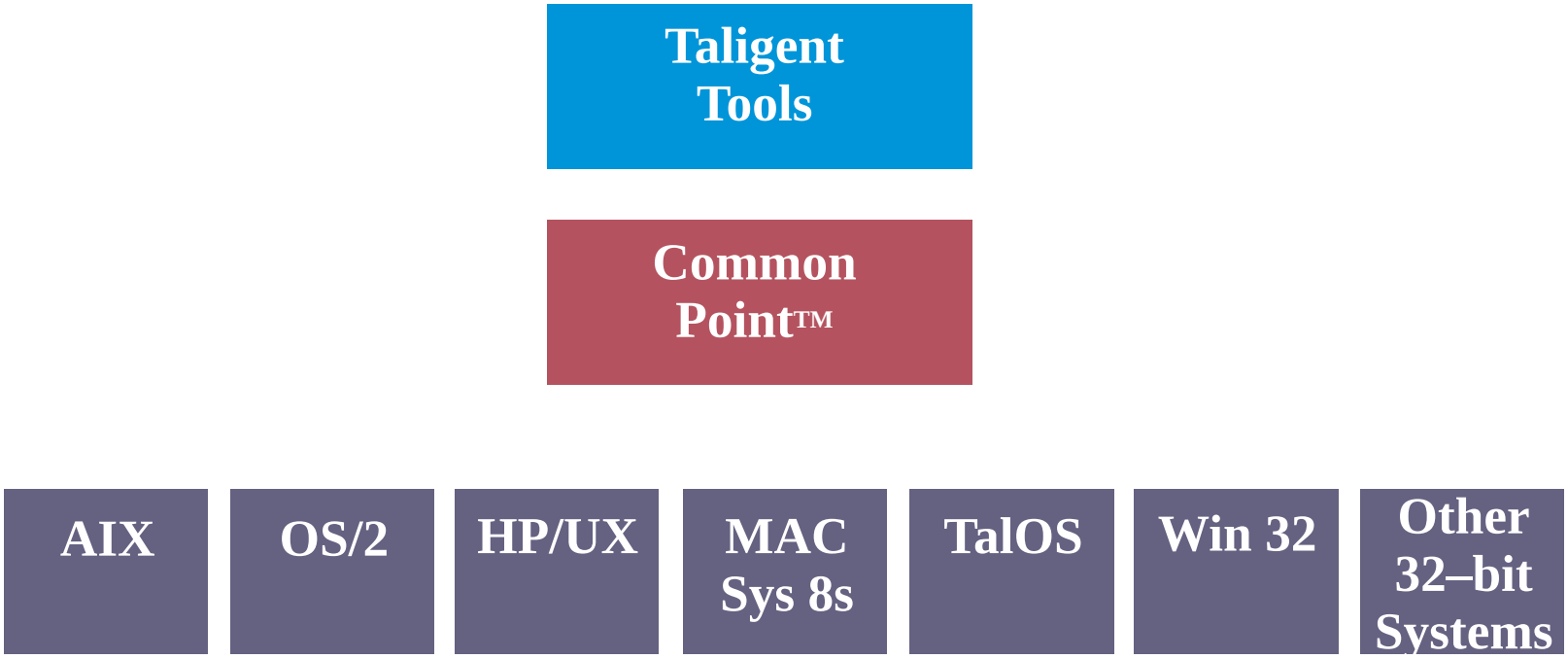
Taligent Development Environment



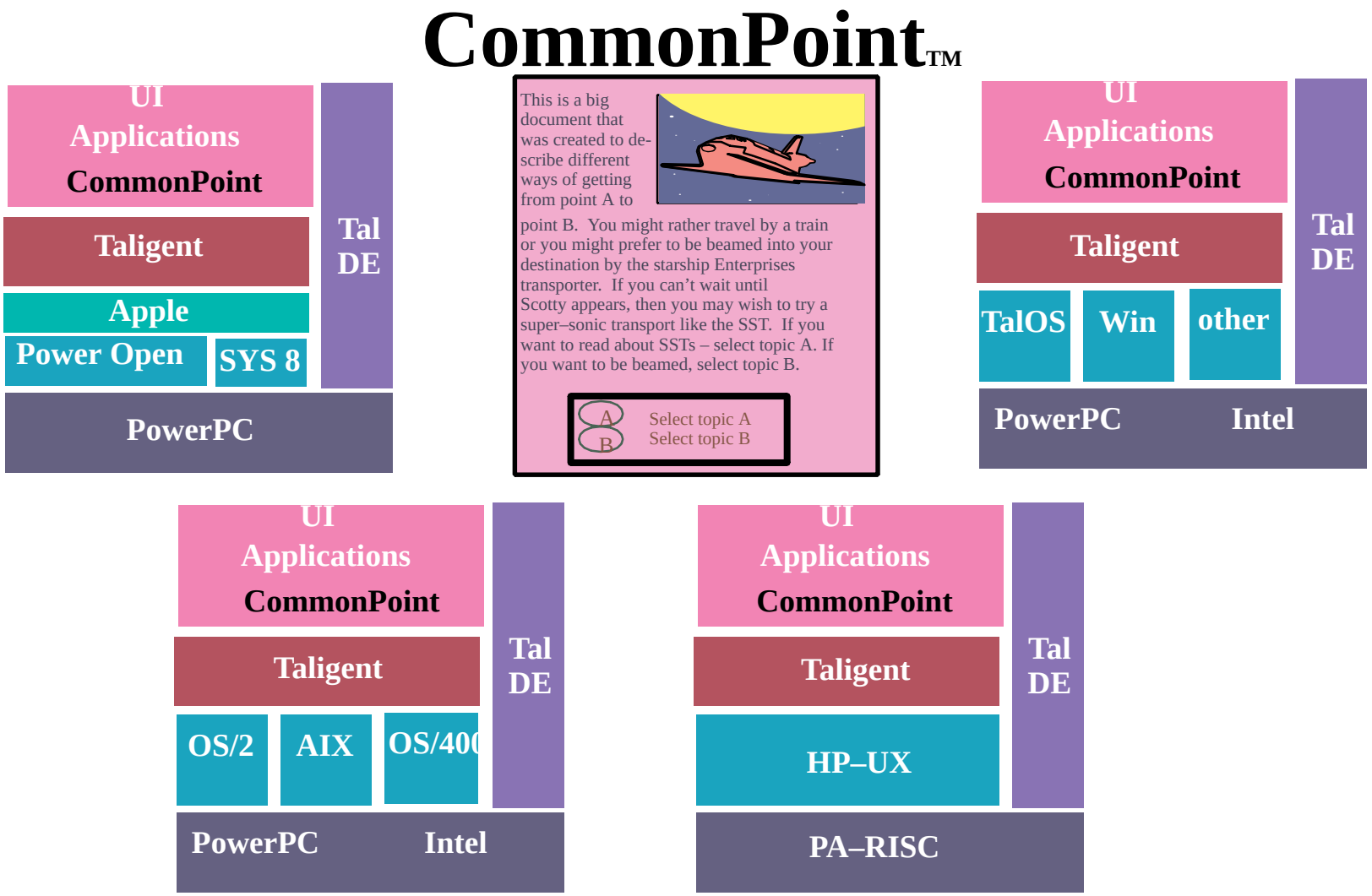
Taligent, ISV, and Investor Tools



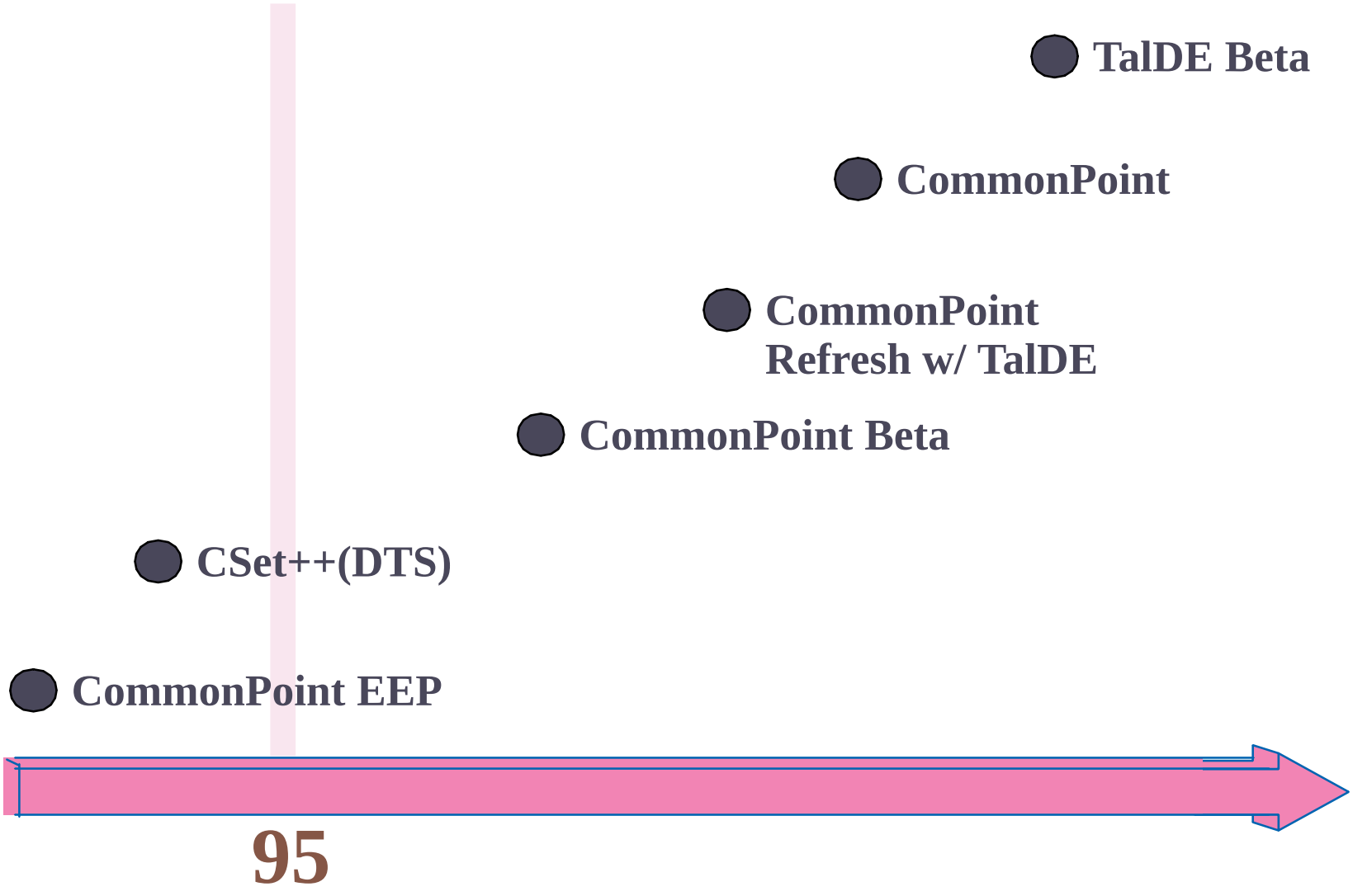
Multiple Platform Development



The Application System Standard



Object technology rollout on AIX



Taligent Value Proposition

- **ISVs/System Integrators**
  - Rapid application development
  - Foundation for innovation
  - High volume deployment
  - New business opportunities
  - A new alternative
- **End–Users**
  - Task–centered computing
  - Rich new application functionality
  - Co–existence and broadest deployment
  - Compelling user experience
- **Corporate IS**
  - Customized solutions
  - Ease of application development and maintenance
  - Full distribute object and client/server support
  - Co–existence and broadest deployment

How to Get Ready

Educate Yourselves
- Learn from peers
- Recruit & manage outside experts
- Invest heavily in training

Evangelize OOT
- Educate mangement about OO
- Recruit a high–level sponsor
- Set expectations appropriately

Organize for OOT
- Target developers based on initiative
- Work in small teams
- Plan for new roles and skills
- Develop incentives to foster reuse

Understand OO Methodologies and the Technology
- Target OO applications wisely
- Understand that anlaysis and design are difficult
- Consider am enterprise model
- Choose an OO language carefully
- Prototype
- Conduct design and code reviews
- Focus on performance issues early
- Properly set expectations for reuse
- Identify OO Metrics